

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 3

Виконав студент ПІ-13 Баран Софія Володимирівна

(шифр, прізвище, ім'я, по батькові)

Перевірів

\_\_\_\_\_  
( прізвище, ім'я, по батькові)

Київ 2021

# Лабораторна робота №3

## Дослідження ітераційних циклічних алгоритмів

**Мета** – дослідити подання операторів повторення дій та набутти практичних навичок їх використання під час складання циклічних програмних специфікацій.

### Варіант 3

З точністю  $\varepsilon=0.00001$  обчислити

$$s = 1 - \frac{x^2 + 1}{3} + \frac{x^4 + 1}{5} - \dots + (-1)^n \cdot \frac{x^{2n} + 1}{2^n + 1} + \dots, \text{ де } 0 < x < 1.$$

#### 1. Постановка задачі:

Обчислити значення суми за допомогою заданої точності  $\varepsilon = 0.00001$ . Обчислення суми ряду з певною точністю  $\varepsilon$  означає, що сума ряду обчислюється до тих пір, поки модуль різниці між поточним і попереднім членом послідовності більше за  $\varepsilon$ . У вигляді формули це твердження можна записати так:  $|((-1)^n * ((x^{2n} + 1) / ((2^n + 1)))| \leq \varepsilon$ . У даному випадку умовою задана рекурсивна формула, прирівняємо її до змінної term, вийде нерівність:  $|term| \leq \varepsilon$ , яка в циклі буде еквівалентною першій нерівності і буде головною умовою заданого ітераційного циклу. Далі потрібно перевірити чи входить x в заданий проміжок  $0 < x \leq \varepsilon$ , розкривши при цьому модуль за допомогою функції abs. Вийде так:  $abs(term) \leq \varepsilon$ . Якщо ні, потрібно ввести коректне значення x. Результатом розв'язку є значення ряду суми при заданому x ( $0 < x < 1$ ).

#### 2. Побудова математичної моделі:

Складемо таблицю імен змінних

Змінна	Тип	Ім'я	Призначення
Задане число $\varepsilon$	Дійсний	$\varepsilon$	Початкові дані
Задане число x	Дійсний	x	Початкові дані
Лічильник n	Цілий	n	Проміжні дані
Змінна term	Дійсний	term	Проміжні дані
Сума s	Дійсний	s	Результат

Якщо  $x \in (0;1)$ , то виконується ітераційний цикл:

1. обчислення  $term$  при заданому  $t$
2. до початкової  $s = 0$  додати  $term$  при заданому  $n$  та порівняти до  $s$
3. збільшемо перемінну  $n$  на одиницю

Повторюємо цикл до тих пір поки  $abs(term)$  не стане  $\geq \epsilon$ , де  $abs$  – функція, яка математично обраховує та відкриває модуль.

Якщо  $x$  не  $\in (0;1)$ , то запросити введення коректного  $x$  ще раз

### 3. Розв'язання:

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію введення змінних  $s$ ,  $x$ ,  $\epsilon$ .

*Крок 3.* Деталізуємо дію належності  $x$  з використанням альтернативної форми вибору.

*Крок 4.* Деталізуємо дію знаходження суми ряду  $s_x$  допомогою ітераційного циклу.

### 4. Псевдокод алгоритму:

*Крок 1*

**початок**

введення  $x$

вводимо змінні  $s$ ,  $n$ ,  $\epsilon$

знаходимо прилежність  $x$  до умови

знаходимо суму ряду  $s$

виведення  $s$

**кінець**

*Крок 2*

**початок**

введення  $x$

$s = 0$ ,  $n = 0$ ,  $\epsilon = 0,00001$

знаходимо прилежність  $x$  до умови

знаходимо суму ряду  $s$

виведення  $s$

**кінець**

### *Крок 3*

#### **початок**

введення  $x$

$s = 0, n = 0, \varepsilon = 0,00001$

**якщо**  $(x > 0)$  та  $(x < 1)$

**то**

знаходимо суму ряду  $s$

**інакше**

виведення «некоректне  $x$ »

**все якщо**

виведення  $s$

**кінець**

### *Крок 4*

#### **початок**

введення  $x$

$s = 0, n = 0, \varepsilon = 0,00001$

**якщо**  $(x > 0)$  та  $(x < 1)$

**то**

**повторити**

$term = (pow((-1), n)) * (((pow(x, 2n)) + 1) / ((pow(2, n)) + 1))$

$s = s + term$

$n = n + 1$

**поки**  $abs(term) \geq \varepsilon$

**все повторити**

**інакше**

виведення «некоректне  $x$ »

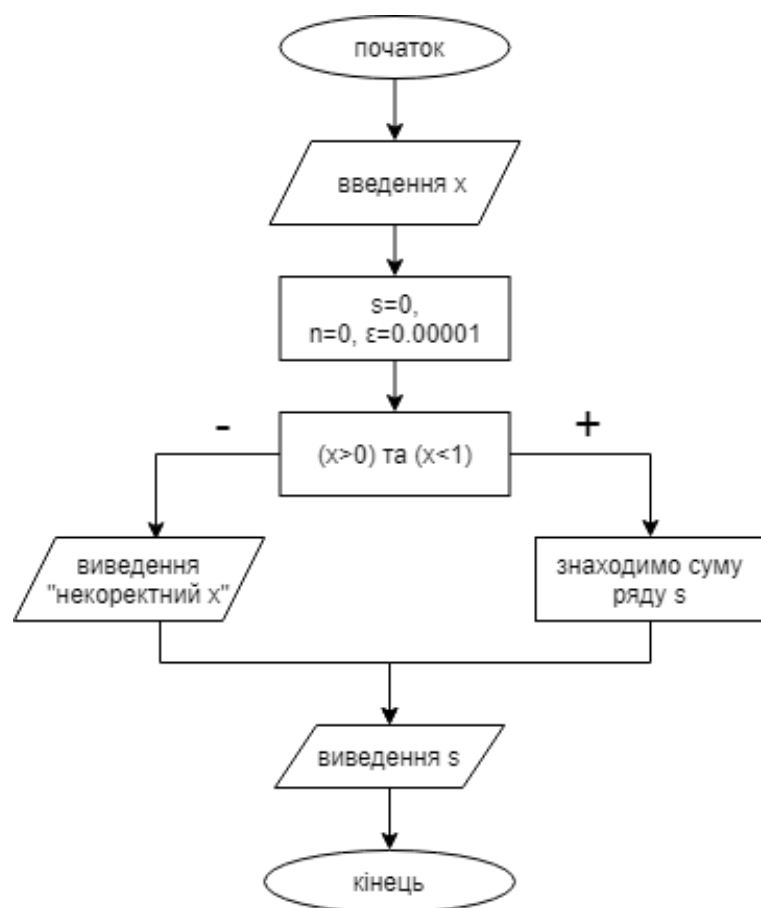
**все якщо**

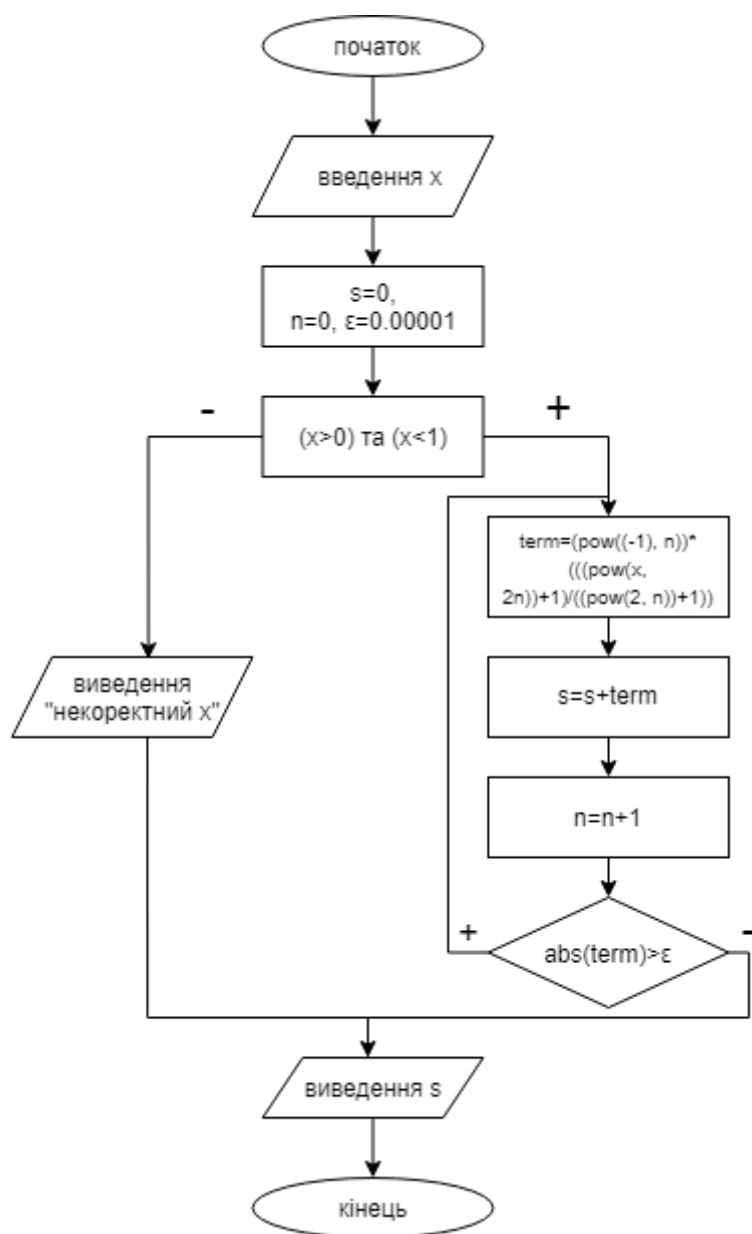
виведення  $s$

**кінець**

## 5. Блок-схеми алгоритму:







## 6. Випробування алгоритму:

Перевіримо правильність алгоритму на довільних значеннях

Розглянемо результат алгоритму при  $x=0.5$

Блок	Дія
	початок
1	введення $x=0,5$
2	введення $s=0, n=0, \varepsilon=0.00001$
3	$0,5>0$ та $0,5<1$ ( $x$ входить в проміжок)
4	$term=1$ $s=1$ $n=1$ $abs(term)$ порівняти з $\varepsilon$ : $1 \geq 0.00001$
5	$term=-0.41666667$ $s=0.58333333$ $n=2$ $abs(term)$ порівняти з $\varepsilon$ : $0.41666667 \geq 0.00001$
	виведення: $0.58333333$
	кінець

Розглянемо результат алгоритму при  $x=2$

Блок	Дія
	початок
1	введення $x=2$
2	введення $s=0, n=0, \varepsilon=0.00001$
3	$2>0$ та $2<1$ ( $x$ не входить в проміжок)
	виведення: «некоректне $n$ »
	кінець

## 7. Висновки

Дослідила подання операторів повторення дій та набула практичних навичок їх використання під час складання циклічних програмних специфікацій. В результаті виконання лабораторної роботи отримала алгоритм для знаходження суми ряду  $s$  при заданній  $x$  за допомогою ітераційного циклу та з врахуванням умови. Побудувала мат. модель, псевдокод, блок схему. Протестувала алгоритм.